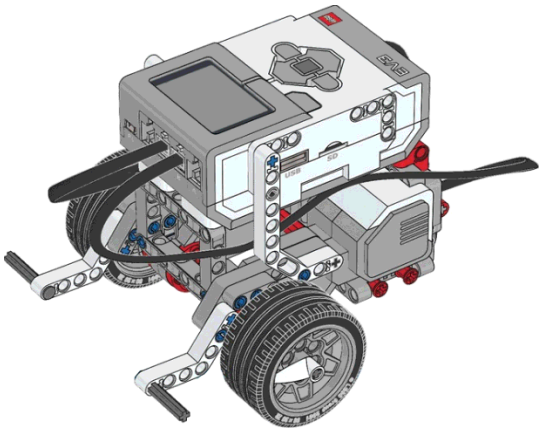
Lunar Rover

Software Requirements Specification

Software Engineering & Project



Team: PG-29
Benjamin Winding
Kin Leong Lee
Pavitterjeet Sidhu
Phan Huy Nguyen
Sean Hennessy
Xiaoshan Chen

20/08/2017

Contents

List of Figures

Revision History

Name	**Date**	**Version**	**Summary of Changes**
Ben	21-Aug-2017	0.1	Initial Draft
Huy Nguyen	24-Aug-2017	0.12	SRS section 4 draft
Ben	27-Aug-2017	0.13	Identificatons numbers added. Grammar improvements. Diagrams revisisted. Assumptions added. Priority classifications Added.
Huy Nguyen	26-Aug-2017	0.14	Revise section 4 - Add requirement id, requirement dependency and requirement priority
Issac	27-Oct-2017	1.0	Release Final version

1 Introduction

1.1 Purpose

This report is to describe the software requirements specification of the lunar rover project. The lunar rover project involves building a prototype rover capable of surveying a designated area automatically. The project is to be completed in semester 2 2017.

1.2 Document Conventions

This document contains a series of user requirements. These requirements have been broken down into categories and given a priority based on the sequence they are stated, the first being the most important feature.

Acronyms are also prevalent in the document. These mainly consist of computer technology acronyms such as:

App Application
GUI Graphical User Interface
SRS Software Requirements Specification
....

1.3 Intended Audience and Reading Suggestions

The intended audience of this document are the Client, this project team and project supervisor. Reading suggestions for this document are:

- Client Specification
- Project Specification
- LEGO©Mindstorms EV3 kit overview
- LeJOS EV3 Java Library overview

1.4 Project Scope

The scope of this project is to demonstrate a prototype for a lunar robot, which is capable of performing an automated survey of a extraterrestrial landscape.

This robot is to be constructed using the EV3 LEGO©Mind-storms robot provided by the client. It is to be controlled via a remote location, but is required to automatically make decisions based on the environment around it.

1.5 References

1. Client Specifications,
2. Project Specifications,
3. Ev3 kit

2 Overall Description

2.1 Product Perspective

The product described in this SRS is a self-contained project. It relies on the existing LEGO©Mindstorms robot, supplied by the client. Previously, the robot cannot be controlled by a human being with software no matter whether manually or automatically. This SRS defines the component of the robot system and the following diagram is identifying the functionalities and interfaces of this system.

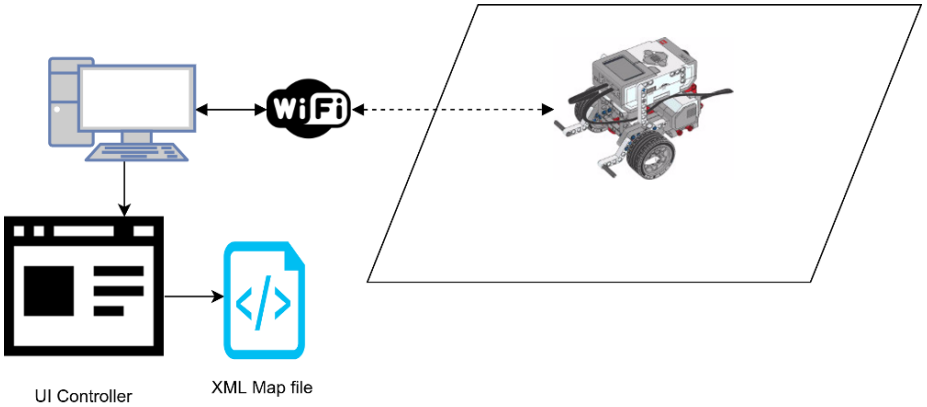


Figure 1: High level system overview

2.2 Product Features

The main features of this robot include

- Automatic survey of specified areas
- Remote control manual override and movement
- On-board obstacle avoidance mechanisms
- No-go zone detection and avoidance
- Ability to return to the starting point or any point selected on mapped area.

2.3 User Classes and Characteristics

The users of the robot include three types; the client, students and teachers. All users must read the user documentation and safety material before operating the robot.

The client is the primary user of the final system. However during development the client may be able to request test runs of the robot, only after reading the relevant user documentation.

Students include project group members and those students enrolled in the university of Adelaide. Students must given explicit permission to use the system by a teacher or project group member.

OS	Minimum Version	Requirements
Windows	7	1 (GHz) or faster 32-bit (x86) or 64-bit (x64) processor*; 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit); 16 GB available hard disk space (32-bit) or 20 GB (64-bit); DirectX 9 graphics device with WDDM 1.0 or higher driver.
Mac	OS X 10	An Intel Core 2 Duo, Core i3, Core i5, Core i7, or Xeon processor; 7 GB of available disk space; 2 GB of RAM.
Ubuntu	16.10	1 (GHz) or faster 32-bit (x86) or 64-bit (x64) processor*; 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit); 16 GB available hard disk space (32-bit) or 20 GB (64-bit);

Figure 2: Detailed operating system requirements

Teachers who have been granted permission to use the system must be familiar with the GUI control and safety precautions prior to using the system.

2.4 Operating Environment

The software can be run on Windows, Mac or Ubuntu systems, specific details are shown in fig 2. WiFi access is required to run the control software. Additionally the GUI must be compiled using JDK (version 1.7 only).

2.5 Design and Implementation Constraints

Implementation Constraints

The system is to be implemented using the EV3 LEGO©Mindstorms kit supplied by the client. The embedded software on the rover is to be written in Java, using the LeJOS Ev3 library version:0.9.0. And the external code used in the system shall not exceed 10%. The build tool for compiling the software and deploying it to the system is *ant*. The system will be implemented using the wireless technology WiFi over an encrypted WPA2 connection.

Development Constraints

The university enterprise Github instance is the version control system to be used throughout the project. During development the project team will use the IntelliJ IDE to design and write code. Slack is used for communication and coordination through out the project. Lastly, Latex is used to produce documentation since it is simple and is easier to maintain in a version control system.

2.6 User Documentation

At the end of the project, a user manual will be available to users via a handbook. The handbook will mainly describe the different parts of GUI, including the

getting started window, the paths and targets browser, the off-line and online browser and so on, assisting users to control the robot better.

2.7 Assumptions and Dependencies

It is assumed that the password is entered correctly and the robot establishes connection successfully with appropriate software and hardware. Also, the robot is ready to be control manually or automatically.

Additional assumptions and dependencies include:

- The robot is controlled in manual mode before in automagical mode
- The volume of battery is enough to support the robot throughout the survey
- Each member of this project are expected to spending 120 hours completing the project
- The robot will not surrounded by the NGZ
- Everyone will complete the assigned sections
- Any area within the map can be landing zones, excluding NGZ's
- The map clearly shows the colours of track/trails
- The map is flat surface
- The map is rectangular
- NGZ's will not be manually entered during the survey.
- Thickness of the line is no less than 1.5cm
- Trail lines are reasonably smooth, no crossing paths

3 User Requirements

3.1 The Map

- **UR01:** The robot shall enter the survey area and produce a survey map
  - **Priority:** High
  - **Dependency:** None
  - **Description:** Upon landing, the robot shall enter the survey area and produce a survey map based on the track followed by the vehicle.
  - **Rationale:** This map will be used to analyze the survey area. This map will also be used to monitor and provide instruction to the vehicle.
  - **Acceptance criteria:** This requirement can be verified by doing a survey of an area by using the software that we are going to develop and the survey map will be automatically produced.

- **UR02:** The map shall be constructed in real time
  - **Priority:** High
  - **Dependency:** UR01
  - **Description:** After landing the robot on the predefined landing zone, the map shall be constructed in real time as long as the robot starts the survey.
  - **Rationale:** This feature will allow the remote operator to monitor and instruct the robot in real time.

  - **Acceptance criteria:** This requirement can be verified by physically observing the robot and comparing the changes on the map.

- **UR03:** The map shall allow to designate NGZ any time
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The remote operate shall be able to designate NGZ any time on the map by using the GUI of the software.
  - **Rationale:** This feature can be very useful in avoid any potential dangerous areas of the map detected by the remote operator.
  - **Acceptance criteria:** This requirement can be verified by drawing a few NGZ on a sample map using GUI.

- **UR04:** The current location of the vehicle shall be visible on map
  - **Priority:** Medium
  - **Dependency:** UR02
  - **Description:** The current location of the vehicle shall be clearly visible on the map in real time. No matter where the robot goes in the map, the operator shall be able to track the current location.
  - **Rationale:** This feature will allow the remote operator to track the current location of the robot.
  - **Acceptance criteria:** This requirement can be verified by moving the robot into different directions and physically comparing the actual position of the robot between the physical map and the map on the GUI.

- **UR05:** The map shall be able to be stored in "XML" format
  - **Priority:** Medium
  - **Dependency:** UR01
  - **Description:** The survey map which shall be created during the process of survey, shall be able to be stored in "XML" format in the system.
  - **Rationale:** This feature will allow the user to store, process, share etc. the survey map.
  - **Acceptance criteria:** This requirement can be verified by storing any sample survey map in "XML" format and then viewing it in "XML" supported platform only.

- **UR06:** The software shall allow to load existing "XML" map file
  - **Priority:** Low
  - **Dependency:** None
  - **Description:** The user shall be able to load any existing partial or fully completed survey map file in "XML" format into the software.
  - **Rationale:** This feature will allow the user to test the robot before sending it to actual survey area and will also be handy in case of system failure.
  - **Acceptance criteria:** This requirement can be verified by loading an exiting "XML" map file in the software and by allowing the robot to survey this map.

- **UR07:** The map shall allow to zoom in on particular area
  - **Priority:** Low
  - **Dependency:** None
  - **Description:** The user shall be able to zoom in on particular area in the map by using the GUI of the software.
  - **Rationale:** This feature can be very useful if the user wants to focus on a particular area on the map and wants to have a closer and detailed look.
  - **Acceptance criteria:** This requirement can be verified by client by zooming in on a particular area of a sample map.

3.2 Sensors

- **UR08:** The robot shall detect craters
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The robot shall detect and avoid going into carters during the survey and try to find an alternate route to destination.
  - **Rationale:** This feature will protect the robot from going into area where it is impossible for robot to recover without assistance.
  - **Acceptance criteria:** This requirement can be verified by trying to send the robot into craters by setting the destination on other side of the carter.

- **UR09:** The robot shall not collide against an external object
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The robot shall detect an external object and avoid colliding with it .
  - **Rationale:** This feature will protect the expensive vehicle from harm and also preserve the integrity of the survey site.
  - **Acceptance criteria:** This requirement can be verified by instructing the robot to move towards an external object.

- **UR10:** The robot shall not go into any NGZ on map
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The robot shall detect and avoid going into any NGZ available on the survey map.
  - **Rationale:** NGZ is considered as potentially dangerous area of the map in which robot shall not go. By doing so, the robot can be protected from any harm.
  - **Acceptance criteria:** This requirement can be verified by trying to send the robot into NGZ by setting the destination of the robot on other side of the NGZ.

- **UR11:** The robot shall be able to detect a track on a given map
  - **Priority:** Medium
  - **Dependency:** None

- **Description:** The robot shall be able to detect and follow any given track on the survey map.
- **Rationale:** This will allow the operator to survey the specific area of the map.
- **Acceptance criteria:** This requirement can be verified by loading an existing survey map with at least one track into software and robot will detect and follow that track.

3.3 Operations

- **UR12:** The robot shall be able to move immediately to a given point
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The remote operator shall be able to place survey point on the map and the robot shall move immediately towards that point.
  - **Rationale:** This feature will allow the operator to survey any area on the map by just setting the survey point on the map.
  - **Acceptance criteria:** This feature can be verified by setting a survey point on the map using the GUI and the robot will move towards it.

- **UR13:** The operator shall be able to stop the robot at any time
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The remote operator shall be able to stop the robot at any time using the stop button on the GUI.
  - **Rationale:** This feature can be useful in protecting the robot or if the operator wants to abort the mission, he/she can just stop the robot and instruct it to return to the landing site.
  - **Acceptance criteria:** This requirement can be verified by pressing the stop button on GUI and in response to this, the robot will stop immediately.

- **UR14:** The software shall provide manual control of the robot as well
  - **Priority:** Medium
  - **Dependency:** None
  - **Description:** The operator shall be able to take manual control of the robot at any time.
  - **Rationale:** This feature can be useful in manually adjusting the position of the robot at any time.
  - **Acceptance criteria:** This requirement can be verified controlling the robot manually at any time using the provided GUI.

- **UR15:** The robot shall return to landing site
  - **Priority:** Medium
  - **Dependency:** None
  - **Description:** The robot shall remember its landing site and shall return to it when the survey is finished.

- **Rationale:** This feature will allow the operator to bring the robot back, once the survey is finished.
- **Acceptance criteria:** This requirement can be verified by instructing the robot to return to the landing site at any time during the survey.

4 System Features

4.1 Map input feature

This feature is designed to allow the operator to manually enter new information to the map such as NGZ, start/destination point, tracks/trails.

Stimulus

1. Open MJ Control application.
2. Enter username/password.
3. Click button "Add map info".
4. Select information type from info type drop-down list (NGZ, tracks/trails etc.)
5. Click on the map to input new map information.
6. Click "Save"/"Cancel" to save or discard map information.

Response

- Save new information enter by operator into map XML file.

Functional requirements

- **FR001**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The application provides a UI to enter new map information.
- **FR002**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The application shall translate new map information into correct XML format and update current XML map info file with new information.

4.2 Map imported/exported feature

This feature designed to allow operator to import a predefine map or export current map information in XML format.

Stimulus

1. Open MJ Control application.
2. Enter username/password.
3. (a) **For import action:** Click "Import map" browse to existed XML map file.

(b) **For export action:** Click" Export map" system will automatically generate latest XML map information with unique name (Map-export-[current-time].xml) in "MapExport" folder of application.

Response

- Load imported map to the UI screen.
- Save latest info to "MapExport" folder when operator click export.

Functional requirements

- **FR003**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The application shall allow operator import initial map information at the beginning of the survey mission. The application shall disable import button while robot is surveying the map.
- **FR004**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The application shall allow operator export map information and save it with a unique name in designated folder.

4.3 Map display feature

This feature is designed to display information of survey area in real-time and robot location. This feature also allow operator to import a XML partial map. The operators can see completed map information with zoom-in/zoom-out supported. Map will display information from imported map, manual input information from operator and real-time information return from robot while it explores the survey area (obstacle, tracks/trail, radiation area etc.)

Stimulus

1. Open MJ Control application.
2. Enter username/password.
3. (optional): Use "import map" button to browse to existing XML predefined map if there is one.
4. (optional): Use manual control mode or autonomous to explore survey area.
5. (optional): Use "Map input" feature to manual add new information to map.
6. (optional): Zoom-in/Zoom-out on a specific region.

Response

- Realtime information displays on GUI map.

Functional requirements

- **FR005**
  - **Priority:** High
  - **Dependency:** None

  - **Description:** The application shall allow to import predefined a xml format map file which comply with predefine DTD provide by client.

- **FR006**
  - **Priority:** Normal
  - **Dependency:** FR005
  - **Description:** The application shall display all information from imported map, operator inputs and survey info return from robot at real-time.

4.4 Manual control feature

This feature designated to allow operator manual control robot. It also has function to switch between manual/autonomous control and emergency stop.

Stimulus

1. Open MJ Control application.
2. Enter username/password.
3. Use navigation button on UI to control robot.
4. Switch between Autonomous/Manual control by turn on/off "Autonomous Control" toggle button.
5. Emergency stop robot by click "Emergency Stop" button

Response

- Robot moving correct when operator use navigation buttons in manual control mode.
- When click on autonomous mode all navigation buttons will be disabled and robot will autonomous survey the area with internal survey algorithm.
- Emergency stop will stop robot intermediately both on manual/autonomous mode.

Functional requirements

- **FR008**
  - **Priority:** High
  - **Dependency:** FR006
  - **Description:** The application shall allow operator manual control robot precisely with navigation buttons.

- **FR009**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** Emergency stop button shall stop robot immediately navigation buttons.

- **FR010**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The application shall allow switch between manual/autonomous mode.

- **FR011**

  - **Priority:** Normal
  - **Dependency:** FR006
  - **Description:** While robot move under manual control. All map information collected from sensor will be recorded and display on map as describe feature 4.3.

4.5 Autonomous control feature

Robot have ability to self-control and autonomously survey the area. Robot also can auto navigate from one point to another point on the map after get instruction from operator.

Stimulus

- Step 1: Open MJ Control application.
- Step 2: Enter username/password.
- Step 3: Step 3: Click the Autonomous Control" toggle button to turn on autonomous mode.
- Step 4 (optional): Click on "Move to Location" button and enter the destination coordinate (x,y). Robot will auto navigate from current location to entered destination.

Response

- Robot will self-control and autonomously survey area then move back to landing point after mission completed. While in autonomous mode all navigation buttons will be disabled.

Functional requirements

- **FR012**
  - **Priority:** High
  - **Dependency:** FR004
  - **Description:** The application shall have an self-control module which can control and navigate robot automatically corresponding to information from the map and sensors data while surveying.
- **FR013**
  - **Priority:** High
  - **Dependency:** None
  - **Description:** The application shall have an obstacle detection algorithms that prevent robot collide with obstacle or move into NGZ's/Craters.
- **FR014**
  - **Priority:** Normal
  - **Dependency:** FR013
  - **Description:** The application shall have a build in algorithm to find the safe and shortest path to navigate from one location to another location.
- **FR011**

4.6 Authentication feature

This feature is designed to prevent un-authorised persons from accessing control of the robot.

Stimulus

1. Open MJ Control application.
2. Enter username/password.

Response

- Grant access to MJ control UI if username/password valid. Otherwise, display un-authorisation message

4.6.1 Functional requirements

- **FR015**
  - **Priority:** Normal
  - **Dependency:** None
  - **Description:** Application shall provide authentication step before operator can start to control robot.

5 External Interface Requirements

5.1 User Interfaces

Graphic User Interface will be provided to allow a user to interact with the system through graphical buttons and map scene. The size of GUI shall be large enough to display system message and map information clearly. This GUI will be implemented by using JavaFX. There are three types of buttons and two types of display area. The specific details of the user interface design will be documented in a separate user interface specification.

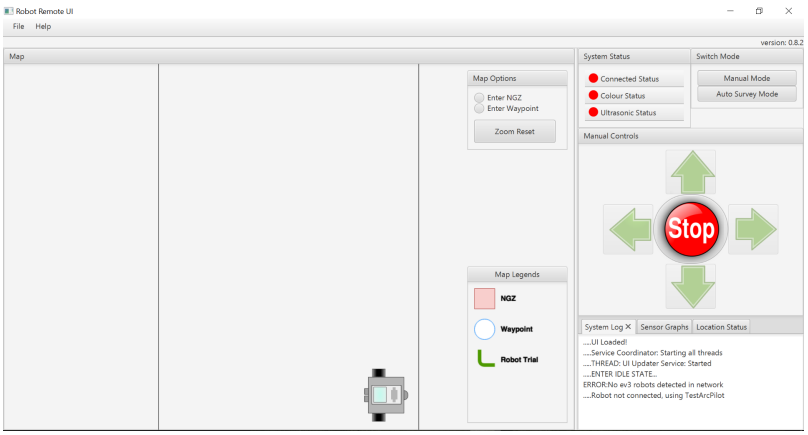


Figure 3: The prototype of GUI

EIR01: Button Functions

Priortiy: High

Dependencies: UR05, UR06, UR14

Description: GUI shall provide a set of buttons for an user to coontrol the robot and perform the features which are described in section 3. The following is the table to illustrate the functionalities of each button.

Button	Keyboard shortcuts	Action
Up	W	Robot moves forward
Left	A	Robot makes a left turn
Right	D	Robot makes a right turn
Down	S	Robot moves backward
Stop	Q	Robot moves backward
Switch Mode	None	Switch the current mode of the robot. If the robot is in Navigation mode then change the robot to Auto mode, and vice versa.
Upload	None	Upload partially complete map data to the system
Download	None	Save the current map information to a local XML file

Figure 4: Buttons functions during manual remote control

EIR02: Map panel

Priortiy: High

Dependencies: UR01, UR02, UR03, UR04

Description: The map information component provides a user interface to view and interact with the map in real time. The following is the table to illustrate the functionalities of map panel.

Button	Action
Mouse Click	Draws a visual NGZ on the map
Scroll wheel	Zooms the map in or out

Figure 5: Buttons functions relating to the survey map

EIR03: User console

Priortiy: Low

Dependencies: None

Description: The User console displays system information and error messages during the operation of the robot. Figure **??**. shows the different messages have been broken into priority categories.

5.2 Hardware Interfaces

There are two main hardware interfaces in the system. The remote control interface is to provide the GUI with an interface to control the robot. The robot embedded interface is to provide the robot with an interface to receive signals from the remote control system.

EIR04:Remote Control Interface

Priortiy: High

Dependencies: None

Priority Level	Message
Level 1: System Status	Connected: Robot, Motors, Sensors
	Disconnected: Robot
	Moving forward
	Moving backward
	Turn left
	Turn right
Level 2: Non-critical	No EV3 found
	No XML found
Level 3: Critical	Fail to connect EV3
	Fail to open motor port
	Fail to open sensor port ¡= System reboot is required
	Fail to load XML file ¡= Please check whether the XML file is created by correct format

Figure 6: Message area levels and message details

Description: This hardware interface is the WiFi which is connected to the computer that contains the control software. The computer must meet the system requirements stated in section 2.4

EIR05:Robot Embedded Interface

Priortiy: Medium

Dependencies: None

Description: All hardware components such as sensors and motors in the Robot are controlled by a controller called a LEGO©EV3 brick. The EV3 brick is also an interface to establish connection between remote system and robot. The communication is established through WiFi. The details of specification of WiFi dongle which is inserted to the Robot will be discussed in 5.4 section.

EIR06:System Control

Priortiy: High

Dependencies: None

Description: The system control is conducted via WiFi connection. The product is an integrated system which is running in remote system only. The robot itself will not contain any software which is required for running the system. The EV3 brick will base all decisions on the received signal from a user remote system to control motors and sensors.

EIR07:Software Interfaces

Priortiy: High

Dependencies: None

Description: The system is developed in Java 8 environment and support running with machine specifications as stated in section 2.4. The flow of software communication between each component is shown in figure 7.

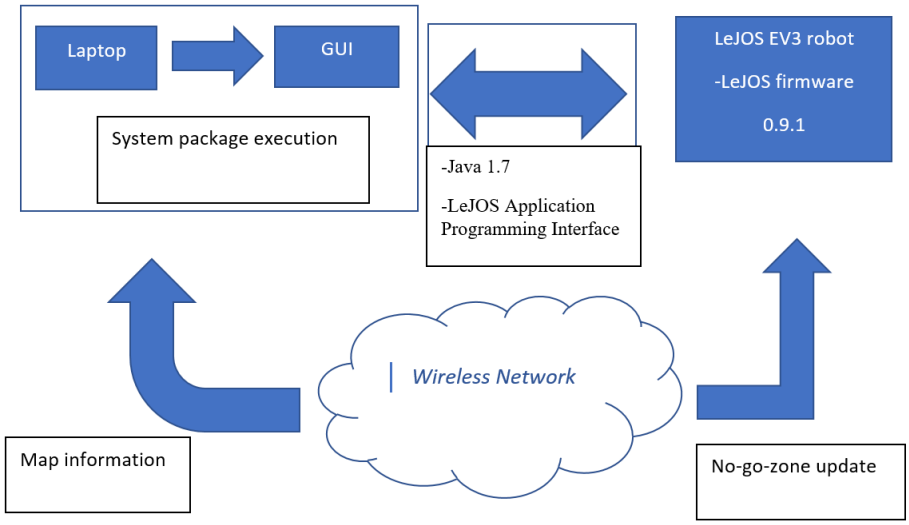


Figure 7: Flow of software communication

Summary of System Package Details

- JDK 1.8 development environment.
- LeJOS 0.9.1 library including ev3 classes.

These two libraries are for making a jar file which can be executed in LeJOS firmware. Also, ev3 classes provide necessary application programming interfaces for the system to communicate with robot.

Summary of Software Components

- LeJOS EV3 is running in Linux and LeJOS EV3 0.9.1 is supporting JAVA 1.8 only. Therefore, the system is developed in JAVA 1.8 environment.
- User must execute the system via provided libraries in the system package. System package contains all necessary libraries for running the system.
- Robot can be controlled by user via GUI only.
- All data and mapping information will be transferred via wireless network. There are 100ms delay. In case of unavailable of wireless network, the system also provides Bluetooth network for communication. However, it is very unstable and is not recommended .

Summary of LeJOS Application Interface

- Manual control
- Package: LeJOS.hardware.motor, LeJOS.hardware.port and LeJOS.hardware.sensor
- This package of API's is for the system to access robot's hardware such motor and sensor
- Automatic control
- Package: LeJOS.robotics.navigation, LeJOS.robotics.pathfinding, LeJOS.robotics.objectdetection, LeJOS.robotics.mapping
- This package of API's is for the system to navigate the map and detect the obstacle automatically. Also, the robot will update the map information to the system

- The detail about mapping mechanism will provide later

5.3 Communications Interfaces

The communication including data transfer and API communication is via wireless network (WiFi). The below is the standard for establishing wireless channel and supporting type of communication.

EIR08:Communication interface between LeJOS ev3 robot and remote system

Priortiy: High

Dependencies: None

Description:

- The communication channel is established via WiFi.
- The Digitech WiFi dongle is used for EV3 to connect wireless network (Support 802.11 b/g/n standards).
- The encryption type for accessing wireless network supports WPA2 or none.

EIR09:The type of communication between LeJOS ev3 robot and remote system

Priortiy: Medium

Dependencies: None

Description:

- All system messages will be transferred to GUI display area only. It will not send any e-mail alerts to user e-mail. For the detail about what will be shown to the GUI display area, please refer to 5.1 user interface.
- Except for software message, the User can get the errors message about EV3 robot via SCP only since there are no external server setting in the EV3 robot.
- Since all functionalities are developed running remotely, all mapping information and initial map are supposed to upload to user's PC. There are no need to transfer any document to EV3 robot.

6 Other Non-Functional Requirements

6.1 Performance Requirements

NFR01: Map Accuracy

Priortiy: High

Dependencies: UR01, UR02, UR03, UR04

Description: The visual representation of the map shall be as accurate as possible. The various objects and hazards that the robot encounters must be recognised and drawn as soon as the robot encounters them.

Rationale: The client must be able to use the map for other purposes once the crash site has been found and therefore is relying on the map to be an accurate representation of the environment. In addition, the robot must be able to use the map to determine where past obstacles were to allow for smooth navigation and to avoid unnecessary travel time.

NFR02: Speed

Priortiy: Medium

Dependencies: None

Description: The surveying of the land and discovery of the goal shall be completed in reasonable time. For this prototype, this shall be interpreted as no more than 25 minutes to complete the journey form landing to returning to the landing zone after completing the goal. On manual control this shall be determined by the operator of the robot but will not be allowed to exceed a specified limit for safety purposes.

Rationale: The robot will have a limited power supply and must be able to complete its mission before that power runs out.

6.2 Safety Requirements

Significant Impact

Priortiy: High

Dependencies: UR09

Description: While on autopilot, the robot shall not exceed $0.3m/s$. When detecting an obstacle, the robot shall stop within 0.15 metres before collision to avoid significant impact and to allow for turning room. During prototype testing, a collision will be deemed significant impact if the robot manages to physically move an obstacle. The manual controller shall default to no more than $0.3m/s$ however the pilot may have additional speed options available to them.

Rationale: The safety of any persons on the landing site is paramount and a collision of the actual robot to any persons may lead to severe injury or death. The prototype is also representing a robot that will be expensive to repair and significant damage to the robot may lead to irreparable damage and failure of the mission.

NFR03: No Go Zones
Priortiy: High
Dependencies: UR10
Description: No more than half the robot may enter the NGZ at any time.

Rationale: An NGZ may represent an area where personnel are active or may indicate an area of known hazardous materials or obstacles otherwise not detectable by the robot. For the safety of the personnel and the robot, the prototype must be able to demonstrate the ability to maneuver its way out of an NGZ without risking injury to personnel or getting stuck in an hazardous area.

6.3 Security Requirements

NFR04: Unauthorised Access
Priortiy: Medium
Dependencies: TBD
Description: The GUI shall have a login section where a username and password is presented and verified before the controller can be used.

Rationale: Unauthorised use of the robot may lead to damage or loss if targeted by malicious competition or an individual who has not been verified by the client.

6.4 Software Quality Attributes

NFR05:Ease of use
Priortiy: Low
Dependencies: None
Description: The GUI shall have intuitive controls and clearly marked buttons. The size of the controller and map shall be set to a size that is sufficient for comfort and legibility. The visual style of the GUI will conform to the standards of similar controller layouts commonly found in maps and remote controllers.

Rationale: The Client will have limited time to spend learning controls and must be able to intuitively navigate the GUI without much training. This should include importing and exporting map data.

7 Other Requirements

8 Appendix A: Glossary

GUI: Graphical User Interface
MJ: "Michael Jackson" - internal nickname to lunar robot and controller ap-

plication
NGZ: No-Go Zone
TBD: To Be Determined
UI: User Interface

9 Appendix B: Analysis Models

10 Appendix C: Issues List